

Score: 0.000 (=0.0%)

Id: 45043

Upišite niz znakova koji nedostaje (tj. označen je s #####) u navedenom CSS kodu?

```
#header h1 {  
  animation-duration: 2s;  
  animation-name: ImeAnimacije;  
  animation-iteration-count: 10;  
  animation-timing-function: linear;  
  animation-direction: alternate;  
}  
  
##### ImeAnimacije {  
  from {  
    margin-left: 100%;  
  }  
  to {  
    margin-left: -100%;  
  }  
}
```

Student's answer:

```
class
```

Hint: Neispravno.

Correct answer:

@keyframes

Score: 1.000 (=100.0%)

Id: 45081

Kako ćemo, slijedeći **BEM** metodologiju, postaviti **crvenu** boju teksta naslova (title) gumba (button)?

a

```
#button-title {  
  color: red;  
}  
...  
<button>  
  <span id="button-title">Button title</span>  
</button>
```

b

```
.btn {  
  color: red;  
}  
...  
<button class="btn">  
  <span>Button title</span>  
</button>
```

c

```
.button--title {  
  color: red;  
}  
...  
<button>  
  <span class="button--title">Button title</span>  
</button>
```

d

```
<button>  
  <span style="color:red;">Button title</span>  
</button>
```

e

```
.btn .title {  
  color: red;  
}  
...  
<button class="btn">  
  <span class="title">Button title</span>  
</button>
```

f

```
.text-danger {  
  color: red;  
}  
...  
<button>  
  <span class="text-danger">Button title</span>  
</button>
```

Score: 1.000 (=100.0%)

Id: 45127

Koja od ponuđenih tvrdnji u vezi SERVER_PUSH mehanizma u HTTP/2 protokolu je ispravna?

- a** Poslužitelj unaprijed šalje tokove podataka klijentu bez HTTP zahtjeva klijenta. Poslužitelj prvo šalje PUSH_PROMISE okvir da bi dojavio namjeru slanja daljnjih tokova. Klijent uvijek prihvaća SERVER_PUSH.
- b** Poslužitelj šalje podatke klijentu i ako PUSH_PROMISE nije zaprimljen.
- c** Poslužitelj uvijek šalje tokove podataka klijentu sa ili bez HTTP zahtjeva klijenta.
- d** Poslužitelj unaprijed šalje tokove podataka klijentu bez HTTP zahtjeva klijenta. Poslužitelj prvo šalje PUSH_PROMISE okvir da bi dojavio namjeru slanja daljnjih tokova. PUSH_PROMISE mora biti zaprimljen da bi se izbjegli višestruki klijentski zahtjevi za istim resursima. Klijent može odbiti SERVER_PUSH slanjem RST_STREAM okvira ako su resursi već u njegovoj privremenoj memoriji.
- e** Poslužitelj u svojoj privremenoj memoriji zapisuje identifikatore resursa koje ima svaki klijent koji mu je ikad poslao HTTP zahtjev.

Score: 0.250 (=25.0%)

Id: 45085

Koje su od sljedećih izjava **istinite**?

- ☒ **a** Kod običnog ("vanilla") CSS-a i klasičnog semantic CSS pristupa tipično moramo pisati sav CSS kod ispočetka ("from scratch") odnosno ne možemo ga ponovo iskoristiti u većoj mjeri za druge projekte
- ☐ **b** Razvoj pomoću radnih okvira (Tailwind, Bootstrap) tipično donosi nepotrebnii višak programskog koda (overhead code)
- ☒ **c** Radni okviri s pomoćnim klasama (utility frameworks, npr. Tailwind) nam omogućuju da slijedimo dobre prakse prilikom razvoja
- ☐ **d** Obični ("vanilla") CSS nam omogućuje brži razvoj nego da koristimo radne okvire
- ☒ **e** Web aplikacije razvijene u sveobuhvatnim radnim okvirima (component frameworks, npr. Bootstrap) imaju nedostatak da "sve izgledaju isto"
- ☐ **f** Obični ("vanilla") CSS nam pomaže da pratimo dobre prakse prilikom razvoja

Score: 1.000 (=100.0%)

Id: 44937

Pretpostavimo da dvije aplikacije (X i Y) koriste uslugu AAI@Edu.hr i protokole OAuth2/OIDC. Prilikom prijave u aplikaciju X i uspješnog unosa korisničkom imena i lozinke na usluzi AAI, prijavljeni smo u aplikaciji X. Nakon toga u istom pregledniku posjetimo aplikaciju Y te se želimo u njoj prijaviti.

Što od navedenog je istina?

- a Prijavom na aplikaciju Y bit ćemo automatski odjavljeni iz aplikacije X
- b Budući da i X i Y koriste AAI@Edu.hr one dijele cookie, pa je cookie aplikacije X odmah prepoznati u aplikaciji Y, čime smo automatski prijavljeni.
- c Aplikacija Y će nas preusmjeriti na uslugu AAI@Edu.hr te ćemo se morati ponovo prijaviti, jer svaka od aplikacija ima vlastiti cookie.
- d Budući da i X i Y koriste AAI@Edu.hr, moramo se prvo odjaviti iz aplikacije X da bi se mogli prijaviti u Y.
- e Aplikacija Y će nas preusmjeriti na uslugu AAI@Edu.hr, ali ćemo odmah biti preusmjereni natrag i biti prijavljeni na aplikaciju Y kao posljedicu činjenice da smo od ranije imali valjani cookie za AAI.**

Score: 1.000 (=100.0%)

Id: 44914

Zamislite sljedeću hipotetsku situaciju:

Napisati ste mobilnu aplikaciju u koju se korisnici prijavljuju koristeći AAI@EduHr, nakon čega aplikacija šalje token (kojeg je izdao AAI@EduHr) Edgaru tražeći od Edgara da vam prikaže trenutni broj bodova na predmetu.

U kontekstu korištenja vanjske usluge za autentifikaciju, što je AAI@EduHr?

a

Resource server

- API koja aplikaciji pruža korisnikove podatke

b

Authorization server

- aplikacija ili usluga na kojoj korisnik ima racun (AAI, Google)

c

Proxy

d

Resource owner

- korisnik na edgaru (student, nastavnik)

e

Client

- aplikacija koja pristupa (Edgar)

Score: -0.250 (=25.0%)

Id: 45059

Kod sigurnosnog nedostatka loša autentifikacija, koji pristup koriste vertikalni napadi:

a cijeli rječnik zaporki za jednog korisnika (npr. sa ili admin)

b slabe zaporce iz rječnika (kratke ili niske entropije)

c brute force

d jedna lozinka za sve korisnike

e napredni automatizirani alati

Score: -0.250 (= -25.0%)

Id: 45051

Koji sigurnosni propust napadač želi iskoristiti ako koristi sljedeću proceduru napada:

1. Napadač kreira hiperlink koji, osim URL-a legitimnog poslužitelja, sadrži i zlonamjerni skriptni kod.
2. Napadač šalje zlonamjerni hiperlink korisniku (npr. putem elektroničke pošte).
3. Korisnik aktivira hiperlink, pri čemu se legitimnom web poslužitelju koji sadrži sigurnosni propust šalje HTTP zahtjev za ranjivom web stranicom.
4. Legitimni web poslužitelj šalje korisniku ranjivu web stranicu kao HTTP odgovor. Zlonamjerni skriptni kod nije umetnut u poslanu web stranicu, nego je još uvijek sadržan samo unutar hiperlinka.
5. Korisnikov web preglednik interpretira ranjivu web stranicu koja se sada nalazi na lokalnom korisnikovom sustavu. Nailaskom na ranjivi dio stranice, aktivira se zlonamjerni skriptni kod iz hiperlinka (kao vrijednost jednog od parametara dobivene web stranice), koji se potom izvršava s ovlastima web preglednika unutar lokalne zone korisnikovog računala.

a Trajni (pohranjeni) XSS sigurnosni propust - sprema se u bazu

b Loša autentifikacija

c Lokalni (DOM) XSS sigurnosni propust

d Jednokratni (reflektirani) XSS sigurnosni propust - hiperlink

e Lažiranje zahtjeva na drugom sjedištu

Score: -0.250 (=-25.0%)

Id: 45071

Koja strategija se preporuča za otklanjanje ranjivosti loša kontrola pristupa?

- a** inverzni Turingov test (CAPTCHA) losa autentifikacija
- b** filtriranje IP adresa losa autentifikacija
- c** sanitizacija unosa od strane korisnika
- d** dodati neku tajnu (token), a ne prihvaćati sve podatke automatski
- e** zamjena javno dostupnih internih referenci s privremenim vrijednostima koje se na poslužitelju preslikavaju u prave

Score: 0.500 (=50.0%)

Id: 45048

Označite sve točne tvrdnje vezane za *XML External Entity (XXE)* sigurnosni nedostatak

- ☐ a Ranjive su sve aplikacije koje parsiraju XML datoteke
- ☐ b U otklanjanju ovog nedostatka potrebno je ograničiti veličinu učitane XML datoteke
- ☒ c Potrebno je izbjegavati korištenje složenijih XML struktura ako nisu potrebne
- ☐ d Učitavanje vanjskih XML datoteka je štetno i potrebno ga je onemogućiti
- ☒ e Potrebno je proučiti i ažurirati postavke XML parsera za učitavanje vanjskih entiteta

Score: 1.000 (=100.0%)

Id: 45057

Web aplikacija za autentifikaciju koristiti sljedeći SQL kôd:

```
String SQLQuery = "SELECT Username FROM Users WHERE Username = '"  
+ username + "' AND Password = '" + password + "'";
```

Koje nizove znakova napadač mora unijeti za vrijednosti varijabli username i password kako bi se mogao prijaviti na sustav kao prvi korisnik u korisničkoj tablici Users?

a

`' OR ''='` i `'`

b

`' OR ''='` i `'+'`

c

`' OR ''='` i `' OR ''='`

d

`' OR ''='` i `' OR ''='`

e

`' OR ''='` i `' OR ''='`

Score: 0.000 (±0.0%)

Id: 45046

Unesi CSS koji će nastati prevođenjem sljedećeg SASS (.scss) koda:

```
%default {  
  color: black;  
}  
%alternative {  
  color: red;  
}  
$brand-color: blue;  
  
h1 {  
  @extend %default;  
  font-size: 2rem;  
}  
  
h2 {  
  @extend %default;  
  font-size: 1rem;  
}
```

<https://beautifytools.com/css-to-scss-converter.php>

(nemojte upisivati komentare, indentacija i novi redovi će biti noramlizirani prije evaluacije)

Ako smatrate da je kod neispravan, upišite "error".

Student's answer:

Hint: Your answer is not correct:

Correct answer:

```
h2, h1 {  
  color: black;  
}  
  
h1 {  
  font-size: 2rem;  
}  
  
h2 {  
  font-size: 1rem;  
}
```

Score: -0.250 (=-25.0%)

Id: 44920

Zamislimo hipotetsku situaciju u kojoj bi za prijavu u Edgar umjesto preusmjeravanja na stranici sustava AAI@Edu.HR, vaše korisničko ime i lozinku za AAI upisali u formu na Edgaru i prepustili Edgaru da obavi prijavu umjesto vas.

Koji OAuth2/OIDC tok bi se koristio?

a Authorization Code Flow + PKCE - aplikacija preusmjeri korisnika na autorizacijski server nakon uspjesne prijave

b Implicit flow - umjesto autorizacijskog koda se vraća pristupni token

c Authorization Code Flow, ali bez PKCE

d Resource Owner Password Flow - korisnik unosi username i password u aplikaciju i onda to prenosi

e Client Credentials Flow =nema korisnika, aplikacija dobije token kroz podatke

Score: -0.250 (=-25.0%)

Id: 44911


Zaglavlje kojim se šalje token poslužitelju ima oblik

a Authentication: Token sadržaj_tokena

b Cookie: token=sadržaj_tokena

c Authorization: Token sadržaj_tokena

d Authentication: Bearer sadržaj_tokena

 **e** Authorization: Bearer sadržaj_tokena

Score: 1.000 (=100.0%)

Id: 45069

Koja strategija se preporuča za otklanjanje ranjivosti lažiranja zahtjeva na drugom sjedištu?

- a** koristiti HTTP POST umjesto HTTP GET
- b** filtriranje IP adresa
- c** sanitizacija unosa od strane korisnika
- d** eliminacija izravnih referenci s privremenim neizravnim vrijednostima
- e** inverzni Turingov test (CAPTCHA)

Score: 1.000 (=100.0%)

Id: 44907

Ako se za provjeru autentičnosti koristi mehanizam *Digest Authentication* korisnik se može odjaviti iz aplikacije

a samo zatvaranjem preglednika

b brisanjem svih kolačića u svom pregledniku

c pozivom postupka na serveru koji će u zaglavlju odgovora imati zaglavlje

`Set-Cookie: digest=invalidate; path=/; expires=Thu, Jan 01 1970 00:00:00 UTC;`

d pozivom postupka na serveru koji će vratiti status 403

e pozivom postupka na serveru koji će u zaglavlju odgovora imati zaglavlje

`Set-Cookie: authentication=null; path=/;`

Score: 0.000 (=0.0%)

Id: 44921

Označite sve vrste tokena koji se koriste u OAuth2/OIDC?

a

code token

b

logout token

c

id token

d

refresh token

e

access token

Score: 1.000 (=100.0%)

Id: 45052

Ako se na poslužitelju ne koriste sigurne reference na objekte pa je osim pristupa putanji:

`/user/getAccounts`

logiranom korisniku bez administratorskih ovlasti dozvoljen pristup putanji:

`/admin/getAccounts`

čime korisnik dobiva administratorske ovlasti, o kojem sigurnosnom nedostaku se radi?

a Nesigurna pohrana osjetljivih podataka

b XSS sigurnosni nedostatak

c Loša autentifikacija

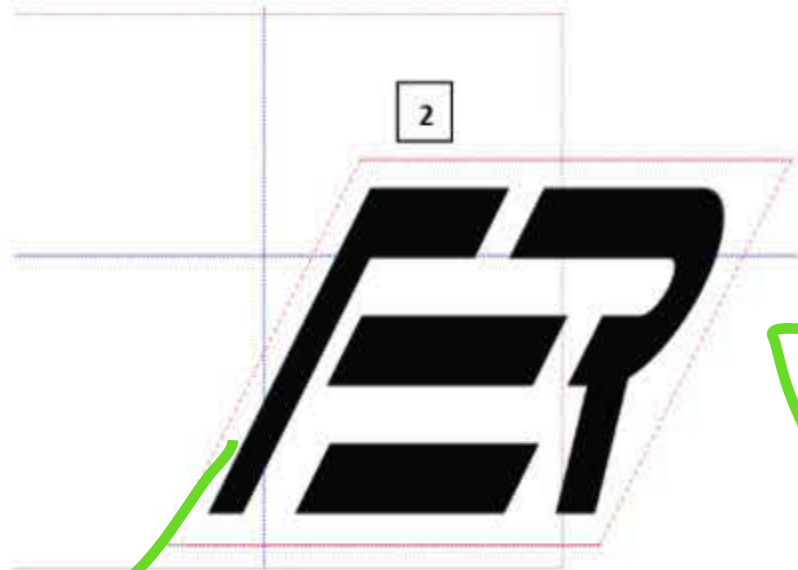
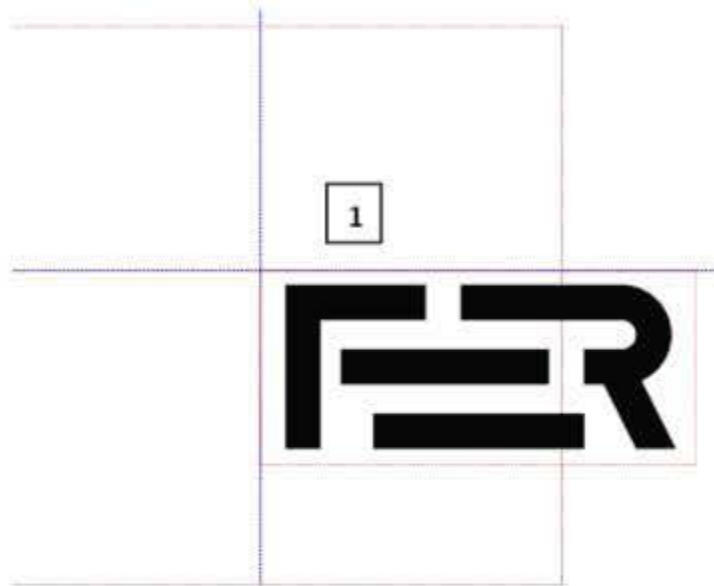
d Lažiranje zahtjeva na drugom sjedištu

e Loša kontrola pristupa

Score: 0.000 (=0.0%)

Id: 45042

Kojim CSS izrazom možemo logo FER-a (sliku) iz inicijalnog stanja 1 dovesti u stanje 2?



1. 45

FER

(45

FER

(0, 45)

FER

(0, -45)

a transform: scale(2, 1) skew(-45deg, 0deg);

b transform: scale(2, 1) skew(45deg, 45deg);

c transform: scale(1, 2) skew(0deg, 45deg);

d transform: scale(1, 2) skew(-45deg, 0deg);

scale(x,y) - skaliraj po x,y osi

skew((za 45 bi FER bio ukosen lijevo po x-,0)

e transform: scale(2, 1) skew(-0deg, -45deg);

f transform: scale(1, 2) skew(45deg, 45deg);

g niti jedan od navedenih izraza

Score: -0.250 (= -25.0%)

Id: 44922

Podaci unutar identifikacijskog token u OpenId Connect protokolu nazivaju se

a permissions

b claims

c restrictions

d scopes

e identities

Score: 1.000 (=100.0%)

Id: 45740

Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi načela jednostavnosti **koje su istinite**:

- ☒ a Jednostavnost se može postići korištenjem TTD (*test-driven development*) metodologije
- ☐ b Potrebno je **predvidjeti** svaki mogući scenarij i svaki rubni uvjet ✓
- ☐ c Jednostavnost se **ne** može postići korištenjem TTD (*test-driven development*) metodologije, štoviše TTD je potrebno uvijek izbjegavati ✓
- ☐ d Načelo jednostavnosti temelji se na korištenju jednostavnih tehnologija na poslužiteljima ✓
- ☒ e Načelo jednostavnosti temelji se na načelima objektno orijentiranog programiranja i cilj je postići lokalnu jednostavnost strukture kôda
- ☒ f Potrebno je **izbjeci predviđanje** svakog mogućeg scenarija i svakog rubnog uvjeta jer se time gubi fokus s najčešćih scenarija

Score: -0.250 (~-25.0%)

Id: 45644

Ako su `Milk` i `Beverage` definirani s

```
type Milk = {
  brand : string;
  fat: number;
  volume: number;
}

type Beverage = {
  type : string;
  brand : string;
  volume : number;
}
```

što će ispisati sljedeći programski odsječak

```
let x : Milk | Beverage = {
  type: 'juice',
  brand : 'Home made',
  volume : 2
};
console.log(typeof x);
console.log(typeof (x as Beverage));
```

- a

Milk | Beverage
Milk | Beverage
- b

object
object
- c

Beverage
Beverage
- d

Milk | Beverage
Beverage
- e

object
Beverage

Score: -0.250 (= -25.0%)

Id: 45664

Koje od sljedećih izjava **su točne**?

a Prilikom iscrtavanja odnosno osvježavanja prikaza, Vue koristi tzv. **Virtual DOM** kako bi optimirao taj proces.

b Vue ne omogućuje dvosmjerno povezivanje (*two way binding*)

c Vue programeru izlaže svoj životni ciklus (*lifecycle events*) na način da postoje funkcije (*lifecycle hooks*) **prije i poslije** svakog od tih događaja, npr. `onBeforeMount()` i `onMounted()`

d Na jednoj stranici može biti samo jedna Vue aplikacija

e Vue nema vlastiti ugrađeni *state management library*, već se najčešće koristi (dodatno instalira) vanjski library koji se zove Vuex.

f Vue ima vlastiti ugrađeni *state management library* koji se zove Vuex.

Score: 1.000 (=100.0%)

Id: 45622

Ako je u Typescriptu napisan sljedeći kod

```
function sum(a, b) {  
  return a + b;  
}
```

kojeg tipa je povratna vrijednosti iz funkcije sum?

a any

b object

c number

d function

e never

Score: 0,000 (=0.0%)

Id: 45742

Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi načela otvoreno-zatvoreno koje su istinite:

a Središnji cilj ovog načela je **potpuno onemogućiti** buduće izmjene softvera i time potpuno ukloniti **sve** troškove promjena

b Ovo načelo odnosi se na stvaranje kôda koji se **mora** mijenjati kada se promijene zahtjevi ili kada se pojave novi obrasci uporabe

c Kôd je **otvoren** za proširenje i **zatvoren** za izmjenu

d Kôd je **zatvoren** za proširenje i **otvoren** za izmjenu

e Ovo načelo odnosi se na stvaranje kôda koji se **ne mora** mijenjati kada se promijene zahtjevi ili kada se pojave novi obrasci uporabe

f Središnji cilj ovog načela je **povećati fleksibilnost** softvera i učiniti buduće promjene **jeftinijima**

Score: 1.000 (=100.0%)

Id: 45739

Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi vertikalnog skaliranja koje su istinite:

a

Vertikalnim skaliranjem povećaju se performanse **postojećih** poslužitelja ili resursa

b

Vertikalnim skaliranjem dodaje se više **istih** poslužitelja ili resursa

c

Kod vertikalnog skaliranja **ne postoji** *scale down* već samo *scale up*

d

Kod vertikalnog skaliranja osim *scale up* postoji i *scale down* jer kvalitetan skalabilan sustav trebao bi omogućiti i smanjenje kapaciteta

e

Vertikalno skaliranje pogodno je za male web aplikacije gdje se kapacitet može održavati **povećanjem** kapaciteta postojećih poslužitelja i veličine već korištenih resursa

f

Vertikalno skaliranje **nije** pogodno za male web aplikacije gdje se kapacitet može održavati povećanjem kapaciteta postojećih poslužitelja i veličine već korištenih resursa

Score: -0.250 (= -25.0%)
Id: 45673
Koje od sljedećih tvrdnji vezane uz *service worker* **su istinite**?

ne ulazi u MI

- a** Service worker se izvodi se mimo glavne preglednikove UI dretve.
- b** Service worker radi samo ako je stranica poslužena putem HTTPS-a ili preko localhosta.
- c** Service worker ne može koristiti *sinkroni* localStorage API, pa zato tipično koristi *asinkroni* IndexedDB.
- d** Kod prvog otvaranja stranice, *service worker* moramo **registrirati** nakon čega preglednik **dohvaća, instalira i aktivira** SW čime on preuzima kontrolu nad *klijentima* u svoj opsegu (*scope*).
- e** Service worker nema pristup DOM-u.
- f** Kada zatvorimo sve kartice/preglednike s domenom/scopem koje je neki service worker kontrolirao, onda se se gasi i prestaje raditi i taj service worker.

Score: 0.000 (=0.0%)

Id: 45674

ne ulazi u MI

Koje od sljedećih tvrdnji **su istinite**?

- a** Dozvolom za notifikaciju, web-aplikacija dobija i implicitnu dozvolu za push notifikacije te ju ne mora posebno tražiti.
- b** Push notifikacije uvelike ovise o preglednicima jer se oslanjaju na push service infrastrukturu u oblaku, koju svaki proizvođač preglednika implementira zasebno
- c** VAPID ključ nam treba kako bi od korisnika zatražili dozvolu za korištenje push notifikacija
- d** Notification API je nezavisan od service workera i može se koristiti i bez njega
- e** Web-aplikacija ne može prikazati notifikaciju bez eksplicitne dozvole korisnika,
- f** Push notifikacija neće biti primljena ako su ugašeni svi prozori preglednika

Score: 0.000 (=0.0%)

Id: 45643

Neka su `Milk`, `T` i `x` definirani kao u nastavku.

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

```
type T = keyof Milk;
```

Označite ispravne naredbe ako je u postavkama TypeScriptovog prevodioca opcija `strictNullChecks` postavljena na `false`.

a `let x : T = "brand" | "fat" | "volume";`

b `let x : T = "brand";`

c `let x : T;`

d `let x : T = "Milk";`

e `let x : T = 1;`

Score: 0.000 (=0.0%)

Id: 45626

Ako su `Milk` i `Beverage` definirani kao u nastavku, što je `Milk & Beverage`?

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

```
type Beverage = {  
  type : string;  
  brand : string;  
  volume : number;  
}
```

a tip koj sadrži sljedeće svojstva

```
{  
  brand : string;  
  fat: number;  
  type : string;  
  volume: number;  
}
```

b `any`

c tip koji u sebi sadrži

```
{  
  brand : string;  
  volume: number;  
}
```

d tip koj sadrži sljedeće svojstva

```
{  
  brand : string[];  
  fat: number;  
  type : string;  
  volume: number[];  
}
```

e tip koji dozvoljava da se u varijablu tog tipa pohrani objekt sa svojstvima identičnim kao u `Milk` ili objekt sa svojstvima identičnim kao u `Beverage`

Score: 0.250 (=25.0%)

Id: 45683

Prilikom **određivanja performansi web sjedišta** označite sve što je potrebno provesti:

ne ulazi MI

a

Procjenu troškova *deploymenta*

b

Test opterećenja (*load test*)

c

Ispitivanje kapaciteta baze podataka (*volume testing*)

d

Ispitivanje responzivnosti (UX)

e

Ispitivanje izdržljivosti (*endurance testing*)

f

Procjenu sigurnosnih ranjivosti

Score: 0.000 (=0.0%)

Id: 45639

Ako je `Milk` definiran kao u nastavku, kako definirati `Beverage` koji za razliku od mlijeka nema `fat`, ali ima svojstvo `type` koje bi predstavljalo naziv vrste pića?

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

a `type Beverage = Milk - {"fat"} + {"type" : string};`

b `type Beverage = Partial<Milk, "fat"> & "type";`

c `type Beverage = {keyof Milk - {"fat"}} & {"type" : string};`

d `type Beverage = Omit<Milk, "fat"> & {"type" : string};`

e `type Beverage = Pick<Milk, "fat"> & {"type" : string};`

a

```
function pickRandomly(data : string) : string;
function pickRandomly<T>(data : string | T[]) : string | T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let s = "Web2";
let c : string = pickRandomly(s);
console.log(c);
```

b

```
function pickRandomly<T>(data : T[]) : T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

function pickRandomly(data : string) : string {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let s = "Web2";
let c : string = pickRandomly(s);
let a = [10, 20, 30, 40];
let n : number = pickRandomly(a);
```

<https://www.typescriptlang.org/play#example/types-vs-interfaces>

c

```
function pickRandomly<T>(data : T[]) : T;
function pickRandomly<T>(data : string | T[]) : string | T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let a = [10, 20, 30, 40];
let n : number = pickRandomly(a);
console.log(n);
```

d

```
function pickRandomly(data : string) : string;
function pickRandomly<T>(data : T[]) : T;
function pickRandomly<T>(data : string | T[]) : string | T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let s = "Web2";
let c : string = pickRandomly(s);
let a = [10, 20, 30, 40];
let n : number = pickRandomly(a);
```

e

```
function pickRandomly<T>(data : string | T[]) : string | T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let s = "Web2";
let c : string = pickRandomly(s);
console.log(c);
```

Score: 1.000 (=100.0%)

Id: 45738

Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi horizontalnog skaliranja **koje su istinite**:

a Kod horizontalnog skaliranja osim *scale out* postoji i *scale in* jer kvalitetan skalabilan sustav trebao bi omogućiti i smanjenje kapaciteta

b Horizontalnim skaliranjem povećaju se performanse **postojećih** strojeva ili resursa

c Horizontalnim skaliranjem dodaje se **više** fizičkih strojeva ili resursa

d Horizontalnim skaliranjem smanjuje se opterećenje na **svakom** stroju ili resursu

e Horizontalnim skaliranjem povećava se opterećenje na **svakom** stroju ili resursu

f Kod horizontalnog skaliranja **ne postoji** *scale in* već samo *scale out*

Score: 1.000 (=100.0%)

Id: 45624

Ako u argumentu funkcije izostavimo tip podatka, TypeScript će smatrati da je taj argument tipa *any*. Ako želimo da takav kod uzrokuje sintaksne pogreške prilikom prevođenja treba uključiti opciju

a `noImplicitAny`

b `resolveJsonModule`

c `noEmitOnError`

d `declaration`

e `strictNullChecks`

Score: -0.250 (=-25.0%)

Id: 45641

Označite odgovor s ispravnim Typescript kodom koji bi omogućio da se sljedeći programski odsječak uspješno prevede i ispiše 5.

```
let p : ImmutablePoint = new ImmutablePoint(2, 3);  
console.log(p.x + p.y);
```

a

```
class ImmutablePoint {  
  readonly x, y;  
  constructor(x: number, y:number) { }  
}
```

b

```
class ImmutablePoint {  
  public readonly x : number;  
  public readonly y : number;  
  constructor(x: number, y:number) { }  
}
```

c

```
class ImmutablePoint {  
  public readonly x : number;  
  public readonly y : number;  
  constructor(public readonly x: number, public readonly y:number) {  
    this.x = x;  
    this.y = y;  
  }  
}
```

d

```
class ImmutablePoint {  
  constructor(public readonly x: number, public readonly y:number) {}  
}
```

e

```
interface ImmutablePoint {  
  constructor(readonly x: number, readonly y:number);  
}
```


Score: 0.000 (=0.0%)

Id: 45653

Što će se ispisati?

```
let obj = { broj: 8 };
let pObj = new Proxy(obj, {
  set: function (obj, prop, value) {
    if (prop === "broj") {
      if (value > 10) value = 10;
    }
    obj[prop] = value;
    return true; // Indicate success
  },
});
console.log(obj.broj);
console.log(pObj.broj);
obj.broj = 15;
console.log(obj.broj);
console.log(pObj.broj);
pObj.broj = 19;
console.log(obj.broj);
console.log(pObj.broj);
```

Student's answer:

```
8
8
15
8
10
19
```

Hint: 8 8 15 8 10 19 <> 8 8 15 15 10 10

Score: -0.250 (=-25.0%)

Id: 45662

Prednosti arhitekture u kojoj koristimo samo **jednostranične web-aplikacije** (u odnosu na druge dvije arhitekture) su:

a

Lagano testiranje

b

Jednostavnost

c

Sadržaj se lako prilagodi klijentu

d

Performanse (dio opterećenja se prebacuje na klijenta)

e

Responzivnost (UX)

f

Sigurnost

Score: -0.250 (=-25.0%)

Id: 45640

Neka je `Milk` definiran kao u nastavku.

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

Pretpostavimo da želimo napisati sljedeći kod

```
let milks : Milks = {};  
milks.DukatMali = {brand: "Dukat", volume: 0.5, fat : 3.2}  
milks.VindijsaVelika = {brand: "Vindijsa", volume: 1.75, fat : 3.2}
```

Što od navedenog treba prethoditi takvom kodu?

a `interface Milks {
 [key:Milk]
};`

b `interface Milks {
 [key:string] : Milk;
};`

c `type Milks = {string : Milk}[];`

d `interface Milks {
 any:Milk
};`

e `type Milks = Milk[];`

Score: -0.250 (=-25.0%)

Id: 45642

Neka su u nekoj TypeScript datoteci zadani sučelje `Point` i funkcija `distance` kao u nastavku

```
interface Point {  
  x:number;  
  y:number;  
}  
  
function distance(p1:Point, p2: Point) : number {  
  return Math.sqrt((p1.x - p2.x) **2 + (p1.y - p2.y) ** 2);  
}
```

Označite sve programske odsječke koji su sintaksno ispravni.

a

```
let d = distance({x:1, y:2}, {x:4, y:6});
```

b

```
const point1 = new Point(1, 2);  
const point2 = new Point(4, 6);  
let d = distance(point1, point2)
```

mora se odrediti konstruktor i ovo mora biti klasa?

c

```
const point1 : Point = {x: 1, y: 2};  
const point2 : Point = new Point(4, 6);  
let d = distance(point1, point2);
```

d

```
const vector = {x : 1, y: 2, z : 3};  
const point = {x : 2, y: 1};  
let d = distance(point, vector);
```

bitno je da postoje x i y, nije vector : Point nego samo objekt

e

```
const vector = {x : 1, y: 2, z : 3};  
let d = distance(vector, vector);
```

Score: 0.000 (=0.0%)

Id: 45760

Kojoj metodologiji odnosno radnom okviru odgovara sljedeći kod:

```
<figure class="md:flex bg-slate-100 rounded-xl p-8 md:p-0 dark:bg-slate-800">
  
  <div class="pt-6 md:p-8 text-center md:text-left space-y-4">
    <blockquote>
      <p class="text-lg font-medium">
        Lorem ipsum...
      </p>
    </blockquote>
  </div>
</figure>
```

a

SMACSS

b

Bootstrap

c

Ništa od navedenog, klasični CSS

d

Utility classes CSS radni okviri poput Tailwinda ili Tachyonsa

e

BEM

f

Foundation

Score: -0.250 (~25.0%)
Id: 45819
Koji isječak programskog kôda je potrebno upisati u **HTML5** dokument **umjesto** XXXXXXXXXXXX da bi se u pregledniku ispravno koristio **Web Storage API** kao na slici?

```
<!DOCTYPE html>
<html>
<head>
<script>
function clickCounter() {
  if (typeof(Storage) !== "undefined") {
    if (XXXXXXXXXXXX.clickcount) {
      XXXXXXXXXXXXXXX.clickcount = Number(XXXXXXXXXXXX.clickcount)+1;
    } else {
      XXXXXXXXXXXXXXX.clickcount = 1;
    }
    document.getElementById("result").innerHTML = "Kliknuli ste na tipku " + XXXXXXXXXXXXXXX.clickcount + " puta u ovoj sesiji.";
  } else {
    document.getElementById("result").innerHTML = "Nažalost vaš preglednik ne podržava HTML Web Storage API.";
  }
}
</script>
</head>
<body>

<p><button onclick="clickCounter()" * type="button">Kliknite ovu tipku</button></p>
<div id="result"></div>
<p>Kliknite tipku kako biste vidjeli povećanje brojača.</p>
<p>Zatvorite karticu preglednika (ili prozor) i pokušajte ponovno, a brojač će se resetirati.</p>

</body>
</html>
```

ne ulazi u MI

Kliknite ovu tipku

Kliknuli ste na tipku 3 puta u ovoj sesiji.

Kliknite tipku kako biste vidjeli povećanje brojača.

Zatvorite karticu preglednika (ili prozor) i pokušajte ponovno, a brojač će se resetirati.

a	navigator.storage
b	document.storage
c	sessionStorage
d	session
e	window.storage

Score: 1.000 (=100.0%)

ne ulazi u MI

Id: 45762

Koje su od sljedećih izjava **istinite**?

a

Obični ("vanilla") CSS nam omogućuje brži razvoj nego da koristimo radne okvire

b

Web aplikacije razvijene u sveobuhvatnim radnim okvirima (*component frameworks*, npr. Bootstrap) imaju nedostatak da "sve izgledaju isto"

c

Kod običnog ("vanilla") CSS-a i klasičnog *semantic CSS* pristupa tipično moramo pisati sav CSS kod ispočetka ("*from scratch*") odnosno ne možemo ga ponovo iskoristiti u većoj mjeri za druge projekte

d

Radni okviri s pomoćnim klasama (*utility frameworks*, npr. Tailwind) proizvode minimalni CSS, te štedimo na veličini CSS datoteka odnosno mrežnom prometu

e

Obični ("vanilla") CSS nam pomaže da pratimo dobre prakse prilikom razvoja

f

Obični ("vanilla") CSS nam omogućuje punu kontrolu nad izgledom sučelja

Score: 0.000 (=0.0%)

Id: 45765

Što će se ispisati?

```
let obj = {
  x: 1,
  y: 2
};
let pObj = new Proxy(obj, {
  get: function (target, prop, receiver) {
    if (prop === "x") {
      return "x is private";
    }
    return Reflect.get(...arguments);
  },
  set: function (obj, prop, value) {
    if (prop !== "y") {
      obj[prop] = value;
    }
    return true;
  },
});
console.log(pObj.x);
console.log(pObj.y);
obj.x = 14;
obj.y = 14;
console.log(pObj.x);
console.log(pObj.y);
pObj.x = 25;
pObj.y = 25;
console.log(pObj.x);
console.log(pObj.y);
// -----
console.log(obj.x);
console.log(obj.y);
```

Student's answer:

```
x is private
2
x is private
14
x is private
25
25
14
```

Hint: x is private 2 x is private 14 x is private 25 25 14 <> x is private 2 x is private 14 x is private 14 25 14

Score: 1.000 (=100.0%)

Id: 45778

Kod izrade **skalabilnih arhitektura aplikacija** označite **sve** tvrdnje u vezi vertikalnog skaliranja **koje su istinite**:

a Vertikalno skaliranje pogodno je za male web aplikacije gdje se kapacitet može održavati **povećanjem** kapaciteta postojećih poslužitelja i veličine već korištenih resursa

b Kod vertikalnog skaliranja **ne postoji** *scale down* već samo *scale up*

c Vertikalno skaliranje **nije** pogodno za male web aplikacije gdje se kapacitet može održavati povećanjem kapaciteta postojećih poslužitelja i veličine već korištenih resursa

d Kod vertikalnog skaliranja osim *scale up* postoji i *scale down* jer kvalitetan skalabilan sustav trebao bi omogućiti i smanjenje kapaciteta

e Vertikalnim skaliranjem dodaje se više **istih** poslužitelja ili resursa

f Vertikalnim skaliranjem povećaju se performanse **postojećih** poslužitelja ili resursa

Score: 1.000 (=100.0%)

Id: 45779

Ako su `MemoryCard` i `Phone` definirani kao u nastavku, odaberite naredbu kojom ćemo u metodi `f` ispisati operacijski sustav mobitela (svojstvo `os` u `Phone`)

```
type MemoryCard = {  
  name: string;  
  storage: {  
    capacity: number;  
    writeSpeed: number;  
    readSpeed: number;  
  }  
}  
  
type Phone = {  
  name: string;  
  storage: {  
    capacity: number;  
    type: string;  
  }  
  os: string;  
}  
  
function f(x: MemoryCard | Phone): void {  
  ...  
}
```

a

```
if (x.os !== undefined)  
  console.log(x.os);
```

b

```
if (x instanceof Phone)  
  console.log(x.os);
```

c

```
if ("os" in x)  
  console.log(x.os);
```

d

```
if (typeof x === typeof Phone)  
  console.log(x.os);
```

e

```
console.log(x.os ?? '');
```

Score: 1.000 (=100.0%)

Id: 45777

Kod izrade **skalabilnih arhitektura aplikacija** označite **sve** tvrdnje u vezi horizontalnog skaliranja **koje su istinite**:

a Horizontalnim skaliranjem smanjuje se opterećenje na **svakom** stroju ili resursu

b Kod horizontalnog skaliranja osim *scale out* postoji i *scale in* jer kvalitetan skalabilan sustav trebao bi omogućiti i smanjenje kapaciteta

c Horizontalnim skaliranjem povećava se opterećenje na **svakom** stroju ili resursu

d Kod horizontalnog skaliranja **ne postoji** *scale in* već samo *scale out*

e Horizontalnim skaliranjem povećaju se performanse **postojećih** strojeva ili resursa

f Horizontalnim skaliranjem dodaje se **više** fizičkih strojeva ili resursa

Score: 0.625 (=62.5%)

Id: 45767

Koje od sljedećih tvrdnji vezane uz PWA tehnologije **su istinite**?

a VAPID ključ nam treba kako bi od korisnika zatražili dozvolu za korištenje push notifikacija

b Service worker se izvodi se mimo glavne preglednikove UI dretve.

c Service worker nema pristup DOM-u,

d Service worker radi samo ako je stranica poslužena putem HTTPS-a ili preko localhosta.

e Service worker ne može koristiti sinkroni localStorage API, pa zato tipično koristi asinkroni IndexedDB.

f Notification API je nezavisan od service workera i može se koristiti i bez njega.

Score: -0.250 (=-25.0%)

Id: 45810

Pretpostavimo da na Edgaru postoji web-servis koji temeljem tokena kojeg izdaje AAI@Edu.HR prepoznaje korisnika te mu vraća popis svih njegovih testova i bodova.

Želite napisati aplikaciju za pametni TV koja će se svakih sat vremena spojiti na taj servis i provjeriti ima li novih podataka, ali ne želite se interaktivno prijavljivati na TV-u, već svoje korisničko ime i lozinku za AAI@Edu.HR pohranite u konfiguracijskoj datoteci vaše aplikacije.

Koji OAuth2/OIDC tok se koristi između vaše aplikacije i servisa AAI@Edu.HR u ovom hipotetskom primjeru?

- ☐ a Authorization Code Flow + PKCE
- ☐ b Detached Credentials flow
- ☒ c Client Credentials Flow
- ☐ d Implicit flow
- ☐ e Resource Owner Password Flow

Score: -0.250 (=-25.0%)

Id: 45817

Koji isječak programskog kôda je potrebno upisati u **HTML5** dokument **umjesto** `XXXXXXXXXXXXXX` da bi se u pregledniku **iscrtao pravokutik** kao na slici?

```
<!DOCTYPE html>
<html>
<body>
  <canvas id="myCanvas" width="300" height="150" style="border:1px solid #d3d3d3;">
    Preglednik ne podržava HTML5 canvas tag</canvas>
  <script>
    var c = document.getElementById("myCanvas");
    XXXXXXXXXXXXXXXX;
    ctx.fillStyle = "#FF0000";
    ctx.fillRect(20, 20, 150, 100);
  </script>
</body>
</html>
```



a var ctx = c.getContext();

b var ctx = c.getContext("3d");

c var ctx = c.loadCanvas();

d var ctx = c.getContext("2d");

e var ctx = c.createCanvas();

Score: -0.250 (=-25.0%)

Id: 45791

Ako je u Typescriptu napisan sljedeći kod

```
function demo(a:number, ...b:number[]) {  
    return [a, ...b];  
}
```

da je a:number, ...b:string[] --> (string | number)[]

kojeg tipa je povratna vrijednosti iz funkcije *demo*?

a any[]

b number | number[]

c object

d any

e number[]

Score: 0.000 (=0.0%)

Id: 45812

Među navedenim označite ključeve koji bi morali biti dio pristupnog tokena po protokolu OAuth2

<div>a</div>	email
<div>b</div>	exp expiration time
<div>c</div>	sub
<div>d</div>	iat issued at
<div>e</div>	code

Score: 0.000 (=0.0%)

Id: 45773

Okvir (*frame*) je najmanji element komunikacije HTTP/2 protokolom. Koje informacije okvir sadržava? Označite **jednu** tvrdnju koja je istinita:

- ☐ a Identifikator toka (*ID*) i identifikator poruke (*Message ID*) kojima okvir pripada
- ☐ b Identifikator poruke (*Message ID*) kojoj okvir pripada
- ☐ c Identifikator toka (*ID*) kojemu okvir pripada
- ☒ d Zaglavlje okvira (*frame header*) i identifikator toka (*ID*) kojemu okvir pripada
- ☐ e Zaglavlje okvira (*frame header*)

Score: -0.250 (= -25.0%)

Id: 45798

Neka je `Milk` definiran kao u nastavku.

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

Pretpostavimo da želimo napisati sljedeći kod

```
let milks : Milks = {};  
let DukatMali = {brand: "Dukat", volume: 0.5, fat : 3.2};  
let VindijaVelika = {brand: "Vindija", volume: 1.75, fat : 3.2};  
milks.DukatMali = 7;  
milks.VindijaVelika = 13;
```

Što od navedenog treba prethoditi takvom kodu?

a

```
interface Milks {  
  [key:string] : number  
};
```

b

```
interface Milks {  
  [key:string] : Milk;  
};
```

c

```
type Milks = number[];
```

d

```
interface Milks {  
  [key:Milk] : number  
};
```

e

```
type Milks = Milk[];
```

Score: 0.000 (=0.0%)
Id: 45763
Unesi CSS koji će nastati prevođenjem sljedećeg SASS (.scss) koda:

```
%default {  
  padding: 1rem;  
}  
$primary-color: blue;  
$secondary-color: red;  
  
header {  
  @extend %default;  
  p {  
    font-size: 2rem;  
    color: $primary-color;  
  }  
}
```

(nemojte upisivati komentare; indentacija i novi redovi će biti noramlizirani prije evaluacije)

Ako smatrate da je kod neispravan, upišite "error".

Student's answer:

```
header p{  
  padding: 1rem;  
  font-size:2rem;  
  color:blue;  
}
```

Hint: Your answer is not correct: headerp{padding:1rem;font-size:2rem;color:blue;}

Correct answer:

This test does not have "show solutions" option enabled.

Score: 1.000 (=100.0%)

Id: 45772

Označite **sve** postupke kojima se otklanja sigurnosni nedostatak kada prilikom učitavanja **XML datoteka** sličnim niže navedenoj dolazi do **preljeva međuspremnika (buffer overflow)** na poslužitelju?

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY>
  <!ENTITY bar "World ">
  <!ENTITY t1 "8bar;&bar;">
  <!ENTITY t2 "&t1;&t1;&t1;&t1;">
  <!ENTITY t3 "&t2;&t2;&t2;&t2;&t2;">
]>
<foo>
  Hello &t3;
</foo>
```

☐ **a** dodati neku tajnu (token) uz *Session ID*, a ne prihvaćati sve XML datoteke automatski

☒ **b** izbjegavati korištenje složenijih XML struktura ako nije baš potrebno

☐ **c** filtriranje dozvoljenih IP adresa i raspona IP adresa

☐ **d** uvijek i potpuno onemogućiti unos svih XML datoteka

☒ **e** napraviti validaciju / sanitizaciju XML dokumenata prije parsiranja

Score: 1.000 (=100.0%)

Id: 45815

Prilikom prijave na sustav e-Građani možete se prijaviti tokenom koje izdaje banke u Hrvatskoj. U kontekstu vanjske usluge za prijavu što je sustav e-Građani?

a

Resource server

b

Resource owner

c

Authorization server

d

Proxy

e

Client

Score: 1.000 (=100.0%)

Id: 45766

Koje od sljedećih izjava **su točne**?

a Prilikom promjene neke od varijabli koje utječu na izgled ekrana, Vue osvježava odnosno ponovo iscrtava cijeli sadržaj ekrana (cijeli DOM).

b *Provide/inject* mehanizam u Vueu nam omogućuje da proslijedimo svojstva niz stablo hijerarhije komponenti, čak i ako komponenta nije neposredni roditelj komponenti koja prima svojstva

c Na jednoj stranici može biti više Vue aplikacija

d Vue ima vlastiti ugrađeni *state management library* koji se zove Vuex.

e Vue ne omogućuje dvosmjerno povezivanje (*two way binding*)

f `actions` u Vuexu su asinkrone

Score: 1.000 (=100.0%)

Id: 45770

ne ulazi

Što je potrebno provesti prilikom **određivanja performansi web sjedišta**? Označite **sve** tvrdnje **koje su istinite**:

a Procjenu **sigurnosnih ranjivosti**

b Ispitivanje kapaciteta baze podataka (*volume testing*)

c Test opterećenja (*load test*)

d Ispitivanje **responzivnosti (UX)**

e Procjenu **troškova deploymenta**

f Ispitivanje **izdržljivosti (endurance testing)**

Score: 1.000 (=100.0%)

Id: 45776

Ako zlonamjerni napadač primjećuje da ima pristup putanji:

`https://www.insecure-web-server-example.com/user?id=10023`

te zbog **nesigurnih referenci na objekte na web poslužitelju** zlonamjerni napadač može pristupiti drugoj putanji:

`https://www.insecure-web-server-example.com/user?id=10024`

o kojem **sigurnosnom nedostaku** se radi?

Označite **jednu** istinitu tvrdnju:

- | | |
|----------|--|
| a | XSS sigurnosni nedostatak |
| b | Nesigurna pohrana osjetljivih podataka |
| c | Loša kontrola pristupa |
| d | Loša autentifikacija |
| e | Lažiranje zahtjeva na drugom sjedištu |